UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/659,975	09/11/2003	Kraig Allan Bottemiller	ROC920030180US1	5041

30206 7590 03/27/2007

IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/659,975	BOTTEMILLER ET AL.	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 January 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-18 have been considered. Claims 1-18 amended as per Applicant's request.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-2, 11 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda et al. (USPN 6,643,803, herein Swoboda), in view of Torrey et al. (USPN 6,145,123, herein Torrey), further in view of Official Notice.

4. As per Claim 1, Swoboda teaches: A method for implementing atomic data tracing in a processor system including an auxiliary processor unit (Column 4, Lines 52-62, the Test/Debug Host) coupled to a central processor unit (CPU) (Figure 1, Target device 10), using the auxiliary processor unit (APU) to perform the steps of:

identifying a trace instruction (Column 4, Lines 59-62);

said trace instruction including a primary op code (Column 2, Lines 15-19) and indicating General Purpose Registers (GPRs) containing information to identify a first GPR containing data to be written into a current trace entry of a trace buffer (Column 4, Lines 59-62, what to trace, for example, memory 0x0 or 0x100), said data to be written

being saved automatically in at least one GPR including said first GPR during normal context switch processing (this is what a context switch is, saving and restoring registers, and Examiner is taking Official Notice that one of ordinary skill in the art would have been motivated to use multithreading (thus context switching) in Swoboda's invention, as there are a multitude of advantages for doing so, such as being able to execute instructions on a long-latency event, or being able to execute multiple processes on a single processor);

signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline (Column 26, Lines 46-48);

signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing (Column 28, Lines 42-43), but fails to teach:

and to identify a set of trace engine registers defining a trace engine to use for said trace instruction; said trace engine including said trace buffer;

said trace buffer and said set of trace engine registers defining said trace engine being accessible by the APU;

checking for an enabled trace engine for said trace instruction,

writing trace data into a trace buffer responsive to an identified enabled trace engine for said trace instruction utilizing said set of trace engine registers defining said trace engine to determine where to write the data into said trace buffer.

While Swoboda teaches a trace instruction, he does not specifically teach a trace engine, nor has it been taught that a check for an enabled trace engine. However, Swoboda does teach that debug data is sent to a test host through a Jtag port.

However, Torrey teaches a system which gets debug information and outputs it to a system through a Jtag port, which uses a trace buffer in order to store the data to be sent out (Column 5, Lines 45-58) to account for the differences in speed between the I/O interface and the system clock. Torrey further teaches that this tracing can be enabled or disabled with a bit, such that the tracing does not need to occur when the user is not debugging the circuit (Column 5, Lines 25-27). Furthermore, it would be required that an instruction would specify where in this trace buffer to write, in order to properly address it, and to be able to read it out. Given the advantage Torrey provides, which is being able to store the trace data to be sent out to the debug host, since it can not be sent out at the same speed at which it is processed, one of ordinary skill in the art at the time the invention was made would have been motivated to include a trace buffer (trace engine) such as Torrey's into Swoboda's invention, along with the other advantageous features disclosed above.

5. As per Claim 2, Torrey teaches: The method for implementing atomic data tracing as recited in claim 1, wherein the step of writing trace data into said trace buffer includes the step of utilizing said set of trace engine registers defining said trace engine consisting of a set of device control registers (DCRs) accessible by the APU to determine where to write said trace data in said trace buffer (Column 6, Lines 5-21. In order to write to a trace buffer, the location of where to write the data in the buffer is required to be stored somewhere).

6. As per Claim 11, Swoboda teaches: An apparatus for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) (Column 4, Lines 52-62, the Test/Debug Host) coupled to a central processor unit (CPU) (Figure 1, Target device 10), said apparatus comprising:

a trace instruction (Column 4, Lines 59-62); said trace instruction including a primary op code (Column 2, Lines 15-19) and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer (Column 4, Lines 59-62, what to trace, for example, memory 0x0 or 0x100);

said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing (this is what a context switch is, saving and restoring registers, and Examiner is taking Official Notice that one of ordinary skill in the art would have been motivated to use multithreading (thus context switching) in Swoboda's invention, as there are a multitude of advantages for doing so, such as being able to execute instructions on a long-latency event, or being able to execute multiple processes on a single processor);

the APU processes said trace instruction performing the steps of signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline (Column 26, Lines 46-48); and signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing (Column 28, Lines 42-43), but fails to teach:

a trace engine; said trace engine including a set of device control registers (DCRs) accessible by the APU, and a trace buffer;

said trace instruction including encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs,

responsive to identifying an enabled trace engine for said trace instruction, and

writing trace data into said trace buffer utilizing said set of device control registers (DCRs) included in said trace engine to determine where to write the data into said trace buffer.

While Swoboda teaches a trace instruction, he does not specifically teach a trace engine containing DCRs and a trace buffer, nor has it been taught that a check for an enabled trace engine is made before writing into said trace buffer. However, Swoboda does teach that debug data is sent to a test host through a Jtag port. However, Torrey teaches a system which gets debug information and outputs it to a system through a Jtag port, which uses a trace buffer in order to store the data to be sent out (Column 5, Lines 45-58) to account for the differences in speed between the I/O interface and the system clock. Torrey further teaches that this tracing can be enabled or disabled with a bit, such that the tracing does not need to occur when the user is not debugging the circuit (Column 5, Lines 25-27). Torrey further teaches that this trace buffer is a circular buffer (Column 5, Lines 65-67) that stores the trace data. Furthermore, it would be required that an instruction would specify where in this trace buffer to write, in order to properly address it, and to be able to read it out, being the equivalent of the index to the DCRs. Given the advantage Torrey provides, which is being able to store the trace data to be sent out to the debug host, since it can not be sent out at the same speed at which it is processed, one of ordinary skill in the art at the time the invention was made would

have been motivated to include a trace buffer such as Torrey's into Swoboda's invention, along with the other advantageous features disclosed above.

7. Claim 15 is substantially similar to Claim 11, and is rejected under 35 U.S.C. 103(a) for the same reasons.

8. Claims 3-6, 8-10, 12-13 and 16-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda and Torrey, further in view of Hoyle et al. (USPN 6,453,405, herein Hoyle).

9. As per Claim 3, Swoboda teaches: The method for implementing atomic data tracing as recited in claim 2, but fails to teach:

wherein said set of device control registers (DCRs) include a trace buffer pointer register for storing a base address of said trace buffer and an offset indicating a current trace buffer entry.

While Swoboda and Torrey disclose a system to trace execution using a trace buffer (requiring addressing provided by the DCRs), the specific method of addressing has not been taught in either reference. However, Hoyle teaches a method to address a circular buffer (which is how the trace buffer in Torrey is set up as), involving a base address, and offset, and a mask (These features are taught in several locations in the reference, but can be seen together in Hoyle's Claim 11, which will be referred to for simplicities sake, but Column 3, Lines 11-14 could also be seen). Given the need to

address the trace buffer in some way, one of ordinary skill in the art would have been motivated to use a method such as the one disclosed by Hoyle, in able to properly address and use the trace buffer as disclosed by Torrey, and as stated in the previous rejection, the system of Swoboda would have needed to include some indication of this information as well to ensure the data was written to the correct location. Therefore, given this need to address the buffer, one of ordinary skill in the art at the time the invention was made would have been motivated to include the teachings of Hoyle in the combination of Swoboda and Torrey.

10. As per Claim 4, Hoyle teaches: The method for implementing atomic data tracing as recited in claim 3 wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register (Claim 11, and Column 3, Lines 11-14).

11. As per Claim 5, Hoyle teaches: The method for implementing atomic data tracing as recited in claim 3 wherein said set of device control registers (DCRs) include a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer (Claim 11).

12. As per Claim 6, Torrey teaches: The method for implementing atomic data tracing as recited in claim 3 wherein said set of device control registers (DCRS) include

a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly (Column 5, Lines 25-28).

13. As per Claim 8, Torrey teaches: The method for implementing atomic data tracing as recited in claim 6, wherein said control register includes a number field indicating a number of bytes to be traced (Column 8, Lines 29-31); and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes (For multiple writes with a larger-than one length trace, the offset would be required to update to continue to point to a valid area in the buffer to write to).

14. As per Claim 9, Torrey teaches: The method for implementing atomic data tracing as recited in claim 3 wherein said trace instruction includes a number field indicating a number of bytes to be traced (Column 8, Lines 29-31); and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes (For multiple writes with a larger-than one length trace, the offset would be required to update to continue to point to a valid area in the buffer to write to).

15. As per Claim 10, Swoboda teaches: The method for implementing atomic data tracing as recited in claim 3 responsive to identifying no enabled trace engine for said

trace instruction, signaling the CPU with said op done signal for allowing the CPU to continue with instruction processing without writing trace data (Column 28, Lines 42-43).

16. As per Claim 12, Swoboda and Torrey teach: An apparatus for implementing atomic data tracing in a processor system as recited in claim 11,

a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly (Torrey, Column 5, Lines 25-28), but fail to teach:

wherein said set of device control registers (DCRs) include a trace buffer pointer register for storing a base address of said trace buffer and an offset into a current trace buffer entry; a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer.

While Swoboda and Torrey disclose a system to trace execution using a trace buffer (requiring addressing provided by the DCRs), the specific method of addressing has not been taught in either reference. However, Hoyle teaches a method to address a circular buffer (which is how the trace buffer in Torrey is set up as), involving a base address, and offset, and a mask (These features are taught in several locations in the reference, but can be seen together in Hoyle's Claim 11, which will be referred to for simplicities sake). Given the need to address the trace buffer in some way, one of ordinary skill in the art would have been motivated to use a method such as the one

disclosed by Hoyle, in able to properly address and use the trace buffer as disclosed by Torrey, and as stated in the previous rejection, the system of Swoboda would have needed to include some indication of this information as well to ensure the data was written to the correct location. Therefore, given this need to address the buffer, one of ordinary skill in the art at the time the invention was made would have been motivated to include the teachings of Hoyle in the combination of Swoboda and Torrey.

17. As per Claim 13, Hoyle teaches: An apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein the APU updates said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer (Column 16, Lines 35-58, the offset is aligned to the proper location for each write).

18. Claims 7, 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda, Torrey, and Hoyle, further in view of DeAngelis et al. (USPN 5,226,153, herein DeAngelis).

19. As per Claim 7, Swoboda teaches: The method for implementing atomic data tracing as recited in claim 6, but fails to teach:

wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and wherein the step of writing trace data into said

trace buffer includes writing a time stamp with said trace data responsive to said control register time stamp value.

While Swoboda teaches the Apparatus as disclosed in the previous claims, Swoboda and the combinations disclosed above remain silent to writing a time stamp with the trace data. However, DeAngelis discloses that a recurring problem in data traces is the inability to relate the data traces to each other in time (Column 1, Lines 37-43). DeAngelis discloses a method to insert a time stamp into traces to remedy this problem (Column 13, Lines 10-28). Furthermore, given that Torrey taught a control register which enabled or disabled traces, the same bits which controlled the trace being enabled would also enable or disable the time stamps as disclosed by DeAngelis. Given the problem of being unable to correlate data, and DeAngelis' solution of using a time stamp in order to overcome this problem, one of ordinary skill in the art at the time the invention was made would have been motivated to include a time stamp in the invention of Swoboda, Torrey, and Hoyle, in order to overcome this limitation.

20. As per Claim 14, Swoboda teaches: An apparatus for implementing atomic data tracing in a processor system as recited in claim 12, but fails to teach:

wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.

While Swoboda teaches the Apparatus as disclosed in the previous claims, Swoboda and the combinations disclosed above remain silent to writing a time stamp

with the trace data. However, DeAngelis discloses that a recurring problem in data traces is the inability to relate the data traces to each other in time (Column 1, Lines 37-43). DeAngelis discloses a method to insert a time stamp into traces to remedy this problem (Column 13, Lines 10-28). Furthermore, given that Torrey taught a control register which enabled or disabled traces, the same bits which controlled the trace being enabled would also enable or disable the time stamps as disclosed by DeAngelis. Given the problem of being unable to correlate data, and DeAngelis' solution of using a time stamp in order to overcome this problem, one of ordinary skill in the art at the time the invention was made would have been motivated to include a time stamp in the invention of Swoboda, Torrey, and Hoyle, in order to overcome this limitation.

21. Claims 16-18 are substantially similar to Claims 12-14, and have been rejected under 35 U.S.C. 103(a) for the same reasons.

Response to Arguments

22. Examiner acknowledges the amendment to Claim 15, and agrees that the 101 rejection has been overcome, as only statutory embodiment are now claimed.

23. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning (in pages 14-15, where Applicant states that the disclosure may not be used as a "road map"), it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction

based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

24. Applicant has made several arguments regarding how the current invention differs from the prior art, however, these limitations are not found in the claims, and appear to be intended use, or advantages listed in the specification which are not made explicit in the claims. As Applicant has not made any specific argument regarding the prior art and the current claims, Examiner will address the new limitations in the claims, with Claim 11 as exemplary (the other added limitations in Claim 1 were previously present in 11).

25. Regarding an op code for the trace instruction, Swoboda teaches that to create a software breakpoint, an instruction is modified to contain a breakpoint by substituting the opcode with a breakpoint opcode, thus Swoboda teaches a trace instruction with an opcode. Swoboda teaches that the user can specify for these breakpoints what data is to be examined, thus the register indicating what data is to be written into the trace buffer. In regards to registers being saved in a context switch, it is common knowledge in the art that registers need to be saved during context switches, in fact, a context switch is the saving and restoring of registers, and context switching is a skill known to

those of ordinary skill in the art for a variety of advantages, and Examiner has taken Official Notice that ones of ordinary skill in the art would use context switching in order to implement multithreading. As for the final new limitation, regarding that the DCRs are used to determine where to write data into the trace buffer, Examiner asserts that in order to use a buffer, there needs to be an indication of where to write data into it. Neither Swoboda more Torrey teach exactly what this method is (which is why Hoyle is brought in for dependant claims), however, some method must exist to address this buffer used by Torrey, as it is inherent that memory can not be accessed without an address or index.

Conclusion

26. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:45-6:15.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100